Faculteit Wiskunde & Informatica | Complexiteit
Technische Universiteit Eindhoven | 21 januari 2004

Opgave 1 Er bestaat géén algoritme dat, gegeven een eindige verzameling domino-typen T (met onbeperkt veel domino's per type) beslist of er wel of niet een T-domino-slang in het boven-halfvlak is van een punt X naar een punt Y. Beargumenteer of de volgende varianten van dit domino-probleem beslisbaar of onbeslisbaar zijn.

a) Gegeven domino-typen T en punten X, Y en Z. Is er een T-domino-slang in het boven-halfvlak van X naar Y en een domino-slang van Y naar Z? (5 ptn.)

b) Gegeven domino-typen T en punten X en Y. Is er een T-domino-slang in het boven-halfvlak van X naar Y van hoogstens 12 stenen? (10 ptn.)

c) Gegeven domino-typen T en punten X en Y. Is er een T-domino-slang in het boven-halfvlak van X naar Y van minstens 12 stenen? (10 ptn.)

Definieer

$$
\begin{aligned}
DS_0 &= \{(X,Y) \mid \text{er is een } T\text{-domino-slang in het boven-halfvlak van } X \text{ naar } Y\} \\
DS_a &= (X,Y,Z) \mid \text{er zijn } T\text{-domino-slangen in het boven-halfvlak van } X \text{ naar } Y \text{ en van } Y \text{ naar } Z\} \\
DS_b &= \{(X,Y) \mid \text{er is een } T\text{-domino-slang in het boven-halfvlak van } X \text{ naar } Y \text{ van lengte} \leq 12\} \\
DS_c &= \{(X,Y) \mid \text{er is een } T\text{-domino-slang in het boven-halfvlak van } X \text{ naar } Y \text{ van lengte} \geq 12\}
\end{aligned}
$$

a) Er is een domino-slang in het boven-halfvlak van X naar Y precies als er een domino-slang in het boven-halfvlak is van X naar Y en van Y naar X. Een beslissingsalgoritme voor DS_a geeft dus een beslissingsalgoritme voor DS_0 (door Z gelijk aan X te nemen). Er bestaat geen beslissingsalgoritme voor DS_0, dus ook niet voor DS_a.

b) Er zijn slechts einig veel domino-slangen in het boven-halfvlak van lengte hoogstens 12 met vertrekpunt X. Enumereer deze systematisch en controleer of ze Y als uitpunt hebben. Zo ja, beslis 'ja'; beslis 'nee' anders.

c) Merk op $DS_0 = DS_b \cup DS_c$. Stel er is een beslissingsalgoritme voor DS_c: beslis 'ja' als $(X,Y) \in DS_b$ of $(X,Y) \in DS_c$; beslis 'nee' anders. Er bestaat geen beslissingsalgoritme voor DS_0, wel voor DS_b, dus niet voor DS_c.

Opgave 2 Geef een deterministische Turing-machine die de taal $\{ a^n b^n c^n \mid n \geq 0 \}$ van woorden met gelijke aantallen a's, b's en c's accepteert. (15 ptn.)

```
# scan op formaat a*b*c*
# en markeer eind met !
d(q0,#) = (q4,!,L)
d(q0,a) = (q0,a,R)
d(q0,b) = (q1,b,R)
d(q1,b) = (q1,b,R)
d(q1,c) = (q2,c,R)
d(q2,c) = (q3,c,R)
d(q3,#) = (q4,!,L)

# markeer begin met !
d(q4,#) = (q5,!,R)
d(q4,a) = (q4,a,L)
d(q4,b) = (q4,b,L)
d(q4,c) = (q4,c,L)
```

```
# vind matchende b
d(q6,a) = (q6,a,R)
d(q6,b) = (q7,#,R)
d(q6,#) = (q6,#,R)

# vind matchende c
d(q7,b) = (q7,b,R)
d(q7,c) = (q8,#,L)
d(q7,#) = (q7,#,R)

# terug naar begin
d(q8,a) = (q8,a,L)
d(q8,b) = (q8,b,L)
d(q8,c) = (q8,c,L)
```

```
# nog een ronde?
d(q5,a) = (q6,#,R)
d(q5,#) = (q5,#,R)
d(q5,!) = (qYES,!,R)

d(q8,#) = (q8,#,L)
d(q8,!) = (q5,!,R)
```

Opgave 3 Het beslissingsprobleem SUBSETSUM is de vraag of er voor een multiset S van natuurlijke getallen en een getal t, een deelverzameling S' van S bestaat zodanig dat $\sum_{s \in S'} s = t$. Het beslissingsprobleem PARTIONING is de vraag of er partitie X_1, X_2 van een multiset X van natuurlijke getallen bestaat zodanig dat $X_1 \cap X_2 = \emptyset$, $X_1 \cup X_2 = X$ en $\sum_{x \in X_1} x = \sum_{x \in X_2} x$.

a) Bewijs dat SUBSETSUM NP-volledig is. (25 ptn.)

b) Bewijs dat PARTIONING NP-volledig is. (10 ptn.)
(Hint: Bewijs SUBSETSUM $\leq_P$ PARTIONING m.b.v. extra elementen $y = A + t$ en $z = 2A - t$ voor geschikt getal A.)

a) Zie Sipser p268–271.

b) Definieer voor SUBSETSUM-instantie $\langle S, t \rangle$, $A = \sum_{s \in S} s$, $y = 2A - t$, $z = A + t$ en $X = S \cup \{y, z\}$. Dit geeft een polynomiale afbeelding $\langle S, t \rangle \mapsto X$.
Stel S heeft een deelverzameling S' met $\sum_{s \in S'} s = t$. Dan heeft X een partitie $X_1 = S' \cup \{y\}$, $X_2 = (S \setminus S') \cup \{z\}$ met gewichten $t + (2A - t) = 2A$ en $(A - t) + (A + t) = 2A$, respectievelijk.
Stel X heeft partitie X_1, X_2. Totaal gewicht van X is $A + (2A - t) + (A + t) = 4A$. De elementen y en z kunnen dus niet samen in X_1 of in X_2 en X_1, X_2 hebben gewicht $2A$. Zonder verlies van algemeenheid, $y \in X_2$. Definieer $S' = \{ s \in S \mid s \in X_2 \}$. Dan heeft S' gewicht $2A - (2A - t) = t$. Dus S heeft een deelverzameling S' van gewicht t.

Opgave 4 Het beslissingsalgoritme TSP vraagt of er, gegeven een verzameling steden S, een afstandsfunctie $d: S \times S \to \mathbb{N}$ en een getal k, al dan niet een rondtoer is met totale afstand hoogstens k. Het beslissingsprobleem metric TSP vraagt hetzelfde voor een afstandsfunctie met de eigenschap dat

$$d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3) \tag{1}$$

voor alle steden $s_1, s_2, s_3 \in S$. Het optimalisatieprobleem voor metric TSP vraagt voor een verzameling steden S en afstandsfunctie d met eigenschap (1) naar de minimiale lengte van een rondtoer.

a) Geef een factor 2 benaderingsalgoritme voor het optimalisatieprobleem voor metric TSP m.b.v. het vinden van een minimal spanning tree. (5 ptn.)

b) Bewijs dat het benaderingsalgoritme gevraagd bij onderdeel a) inderdaad een factor 2 benaderingsalgoritme is. (10 ptn.)

c) Bewijs dat er geen benaderingsalgoritme voor het optimalisatieprobleem voor TSP bestaat, tenzij P=NP. (10 ptn.)
(Hint: Maak voor benaderingsfactor α en een graaf $G = (V, E)$ een afstandsfunctie $d: V \times V \to \mathbb{N}$ met $d(u, v) = \alpha \cdot \#V + 1$ als $(u, v) \notin E$. Relateer het bestaan van een Hamilton-circuit in G aan het vinden van een benaderende rondtoer van waarde $\alpha \cdot \#V$.)

a) Gegeven een verzameling steden S met afstandstabel d, construeer de minimal spanning tree T voor S m.b.t. d. De graaf T' is de boom T maar dan met verdubbelde kanten. Construeer een Euler-pad E' in T'. Doorloop E, maar sla punten die al geweest zijn over. Dit geeft de rondtoer. Het gewicht *appr* van de rondtoer is de benaderende waarde.

b) Het vinden van een minimal spanning tree en van een Euler-pad in een graaf is polynomiaal. Het algoritme gegeven bij onderdeel a) daarom ook. De lengte van het Euler-pad E is $2M$ met M het gewicht van T. Omdat $d(s_1, s_n) \leq d(s_1, s_2) + \cdot + d(s_{n-1}, s_n)$ is het uitgedunde pad niet langer, dus $appr \leq 2M$.
Laat *opt* het gewicht zijn van de minimale rondtoer. Een minimale rondtoer is een spanning tree voor S met een kant extra, dus $opt \geq M$ en $2M \leq 2opt$. Hieruit volgt voor de benaderende waarde *appr* dat $appr \leq 2M \leq 2opt$.

c) Laat α de factor voor het benaderingsalgoritme zijn. Dus voor instantie (S, d) met benaderende waarde *appr* en optimale waarde *opt* geldt $appr \leq \alpha \cdot opt$. Definieer voor een graaf $G = (V, E)$ de afstandsfunctie $d\colon V \times V \to \mathbb{N}$ door $d(u, v) = 0 als$u=v,d(u,v) $= 1$ als $u \neq v, (u, v) \in E$, $d(u, vo) = \alpha \cdot \#V + 1$ als $u \neq v, (u, v) \notin E$.
Als G een Hamilton-circuit heeft, dan is de optimale TSP-waarde voor (V, d) gelijk aan $\#V$. Dus $appr \leq \alpha \cdot \#V$.
Als $appr \leq \alpha \cdot \#V$, dan heeft de rondtoer geen verbinding van gewicht $\alpha \cdot \#V + 1$, m.a.w. de rondtoer gebruikt alleen kanten van G. Dus G heeft een Hamilton-circuit.
Er geldt dus G heeft een Hamilton-circuit precies als de benaderende waarde is hoogstens $\alpha \cdot \#V$. We hebben dan een polynomiaal beslissingsalgoritme voor Hamilton-circuits: berekende *appr* en geef 'JA' als $appr \leq \alpha \cdot \#V$ en 'NEE' anders. Zo'n algoritme bestaat niet, tenzij $P = NP$.